Seeking A Foundation-Building SRE/Systems Administrator

Do you want to influence the tools and processes of hundreds of SREs who collectively manage tens of thousands of machines?

My team builds, maintains, and manages the back-end systems that are behind the back-end systems that are behind the biggest Apple services, such as iCloud, iTunes, the App Store, and Maps. That wasn't a typo; we build the systems that build the machines that run those services. We have direct influence over more than a hundred thousand bare metal boxes scattered all over the world, almost all running Linux (and almost all are OEL specifically). Our customers are back-end SREs, both for the services mentioned and several others. Our work typically starts when the system is racked and on the network, and ends when it's ready to run user-facing software.

(Hi! I'm the actual hiring manager for this position. You can reach me at robin_powell+jpaug2018@apple.com , and you should feel free to do so. Please feel free to share this job description with qualified people you know. This is my personal work email address, so I ask that it not be shared on the general internet or with recruiters.)

Exactly what you'll end up doing in this role is extremely flexible, because the things other teams need are always changing, but here's some things you, as a great candidate, should expect to happen in your first year:

- You will have taken charge of one of our major systems, becoming the subject matter expert. These systems include, but our not limited to:
    - Our machine provisioning system (homegrown on top of Kickstart) and attendant low-level systems (i.e. DHCP and PXE troubleshooting)
    - Our software packaging and package distribution system (homegrown on top of mrepo and Jenkins)
    - Our login/directory system (aka LDAP)
    - Our monitoring and alerting system (entirely homegrown; alerting has a mild resemblance to Nagios) and making sure that all alerts are actionable
    - Our DNS system, which pulls data from our homegrown CMDB in extremely complex ways
    - Front-line support (we all do this, but having someone with that as their primary job is helpful)
- You will have become known to our customers, and to your peers, for your expertise in your chosen area.
- You will have kept a keen eye on security issues in every project you work on, and you will have contributed to improving security in the systems that were already in place.
- You will have contributed great code to our configuration management system (Puppet on top of a homegrown CMDB) or any of our many automation systems

(which are mostly Perl and Ruby, but also some Python and a bit of Go, and of course tons of shell scripting (Bash) to hold it together).

- You will have become known to our customers, and to your peers, for your helpfulness, ability and willingness to teach and mentor others, and friendly demeanor.
- You will have become known to our customers, and to your peers, for your expertise at debugging and fixing operational issues, such as problems with system configuration, system provisioning, and user access.
- You will have successfully influenced people on other teams to adopt our tools, or improve their automation, or any number of other persuasions that will make their lives better and reduce toil.
- You will have participated in our on-call rotation (which, honestly, is pretty light as these things go; maybe 3 off-hours events per week) and will have contributed to making it better and reducing the toil associated with it.
- You will have actively participated in many, many discussions inside our team and with other teams designed to identify and pursue the best solutions to our automation and systems management problems. You will have brought to these discussions your strong opinions and respectful, collaborative attitude.
- You might have also established a particularly strong relationship with a single other team, such as the monitoring team, the security design team, or one of the property SRE teams. This will have allowed you both to influence them more effectively in their pursuit of automation and toil reduction, and to keep the rest of our team apprised of upcoming initiatives that we need to know about.
- You might have also taken on the challenge of writing an entirely new piece of automation, including customer-facing documentation, operational documentation, extensive automated testing, operational design, release and deployment. There's not a lot of room in our environment for entirely new pieces of automation, though, so this doesn't happen often.